Error-driven learning for self-calibration in a neuromorphic path integration system

Kreiser, Raphaela ; Renner, Alpha ; Sandamirskaya, Yulia

Error-driven learning for self-calibration in a neuromorphic path integration system

Raphaela Kreiser
Institute of Neuroinformatics,
University of Zurich and ETH Zurich

Alpha Renner
Institute of Neuroinformatics,
University of Zurich and ETH Zurich

Yulia Sandamirskaya
Institute of Neuroinformatics,
University of Zurich and ETH Zurich

Abstract

Neuromorphic hardware offers a computing substrate that matches the neuronal network-based robotic controllers. To avoid a bottleneck between neuromorphic hardware and a conventional computer, we aim to realize the whole perception-action control loop on the neuromorphic device. In this work, we show an example of such neuromorphic architecture for one of the basic capabilities of a mobile autonomous agent: estimation of the current pose of the agent based on the executed movements using path integration. Accurate path integration requires alignment between the integration speed in the internal pose-representation and the actual pose change. In this paper we propose an Spiking Neural Network (SNN) for state estimation in 1D, inspired by heading direction networks found in animals. Further, we introduce a mechanism for autonomous calibration of the path integration system. The network adapts the integration speed to the rotational speed of the agent if an error in pose estimation is detected at a loop closure event. We implement and validate the network on a low-power spiking neuromorphic processor Loihi. This implementation of autonomous calibration in an on-chip SNN is an important component of efficient and adaptive state estimation and map formation in a purely neuromorphic robotic controller.

1 Introduction

Foraging in unknown environments is a crucial ability for the survival of most animals and is a basic requirement for mobile robots. Both animals and robots use available sensory information from different modalities in order to estimate their orientation and position with respect to a reference point and direction. Using self-motion cues (odometry) to achieve this is called path integration and becomes particularly important when visual information is not available, ambiguous, or unreliable [*Cheung*, 2014]. In robots, odometry can lead to large error accumulation due to wheel slippage, uneven terrain and sensor errors. In humans and animals, the estimated position drifts with increasing uncertainty when navigating in the dark [*Cheung et al.*, 2012]. However, even simple animals are capable to reliably navigate in unknown environments with computational and energy efficiency yet unmatched by technical solutions. Drawing inspiration from biology can thus reveal promising insights, and first approaches toward biologically inspired localization and mapping have been proposed [*Milford et al.*, 2004; *Tang et al.*, 2018]. Head direction cells, grid cells, time, place, and speed cells are only some of the selective neurons and circuits that have been discovered in the hippocampus and entorhinal cortex (EC) of mice, rats, and bats and seem to contribute to navigation-related capabilities. Neural networks composed of these different neurons enable the animal to consistently integrate information over space and time in order to estimate its location, build a map, and navigate in the environment. Plasticity in the neural circuitry enables perpetual adaptation and calibration of the navigational system, leading to reliable and reproducible performance despite of noise and uncertainties in the sensory systems.

Use of such biologically inspired circuits in robotics is limited by the computational overhead of simulating the neuronal dynamics in software, which makes it not suitable for real-time processing. Neuromorphic hardware, to the contrary, offers a physical computational substrate for directly emulating the biophysics of neurons and synapses in real time in electronic circuits of VLSI[1] devices [*Indiveri et al.*, 2009; *Benjamin et al.*, 2014; *Chicca et al.*, 2014], enabling low latency and high efficiency through massively parallel event-based computation.

Appearing in Proceedings of the Workshop on Robust Artificial Intelligence for Neurorobotics (RAI-NR) 2019, University of Edinburgh, Edinburgh, United Kingdom.

[1] Very Large Scale Integration

In this work, we focus on one of the subsystems of the state-estimation network – the Head Direction network (HD). The HD Network is well established in the fruit fly's ellipsoid body (EB), where a topographical ring of neurons forms a continuous attractor network with an activity bump that is moved in a compass like manner by left or right shifted input from the protocerebral bridge (PB) [*Turner-Evans et al.*, 2017; *Seelig and Jayaraman*, 2015]. A similar non-topographical mechanism was also proposed for the mammal HD System [*Skaggs et al.*, 1995; *McNaughton et al.*, 2006]. The HD bump is driven by the landmark-based visual input, however, in its absence, the HD network integrates self-motion cues [*Seelig and Jayaraman*, 2015]. This means that the neural representation of angular velocity has to be mapped to a specific activation level of the shift-inducing neurons (in the protocerebral bridge in the fly) to move the bump with the correct speed. Due to the availability of both modalities (visual and odometric) at most times, an adaptive calibration of the speed input is possible and seems more likely than a hard-wired mapping. In this work, we propose such an adaptive mechanism that uses the difference between integrated and true (visual) orientation estimate as an error signal to adjust the speed input that shifts the bump.

We build our model on a recently introduced neuromorphic implementation of the HD network [*Kreiser et al.*, 2018]. In that work, the recurrent connectivity in a population of heading direction (HD) neurons gives rise to a localized activity bump on a ring-attractor network, which is shifted in the robot's turning direction driven by the velocity input. In order to correct the odometric drift, a visual cue from the robot's visual sensor is used to reset the estimated HD orientation activity to the "correct" orientation, while the integration speed in the HD network is kept constant and is not corrected. In this work, we propose to augment the HD network with an adaptive mechanism that uses the information of the visual cue not only to reset the estimated orientation, but also to correct the integration speed of the HD network, to achieve more precise state estimation during subsequent path integration.

We first show how by controlling the firing rate of input to the HD network, the integration speed of the velocity commands (i.e. the speed with which the activity bump moves) can be adjusted. Second, we implement an error estimating network that can increase or decrease the firing rate of the input neurons in accordance with the error sign and magnitude. Finally, we show how different angular velocities can be set in the HD network reliably. We demonstrate functionality of the SNN on the neuromorphic computing device Loihi, driven by simulated robotic data.

2 Materials and Methods

2.1 Neuromorphic Hardware: Loihi

An overview of spiking neuromorphic processors can be found in [*Thakur et al.*, 2018]. In this work, we realized the SNN architecture on Intel's neuromorphic research test chip Loihi [*Davies et al.*, 2018]. Loihi uses an asynchronous digital design to implement event-driven parallel computations. The processor implements on-line learning and inference with high efficiency. The chip consists of 128 cores realizing up to 130,000 artificial neurons and features a unique programmable microcode learning engine for on-chip SNN training.

2.2 Neural Network Architecture

Fig. 1 illustrates the schematics of the network architecture. Neurons in the HD population correspond to 360° of HD orientation and form a ring with a localized-bump attractor dynamics. Two neuronal populations – the shift right (SR) and shift left (SL) neurons – form shift layers that receive excitatory input from two Angular Velocity (AV) neurons. One of these neurons is driven by a Clockwise (CW) motor signal, the second one by a Counter-Clockwise (CCW) motor signal through a set of plastic synapses. The AV neurons can drive the shifting neurons with different firing rates. Shifting neurons are connected to another ring of neurons – integrated heading direction (IHD) population – with a shift relative to a one-to-one connectivity. Thus, the activity bump observed on the HD ring is shifted on the IHD ring. Finally, the one-to-one connectivity from IHD to HD updates the activity bump on the HD ring, inducing its movement. In this work, we use the property of this circuit to shift the activity bump in the HD population faster if the firing rate of the shifting layers is higher. The learning circuit that we describe next adjusts this firing rate to remove the state estimation error.

The right half of the network in Fig. 1 shows the error estimation network. This network contains a 2D layer of neurons that receive input from the HD neurons and from the "correct" HD neurons. These two 1D inputs enter the 2D network along different dimensions and are integrated by the neurons in the error-estimation network. Neurons, for which the two 1D activity bumps overlap, are activated, leading to activation in a single region of the 2D error-estimating population. Neurons in the error estimation network are connected to an output layer that consists of two populations, representing the positive or the negative error. The position of the active region in one of the output populations corresponds to the difference between the positions of the bumps in the HD and the "correct" HD populations. For a negative error, the input firing rate to the HD population needs to be decreased to slow down the activity bump, while for a positive error, the input firing rate needs to be

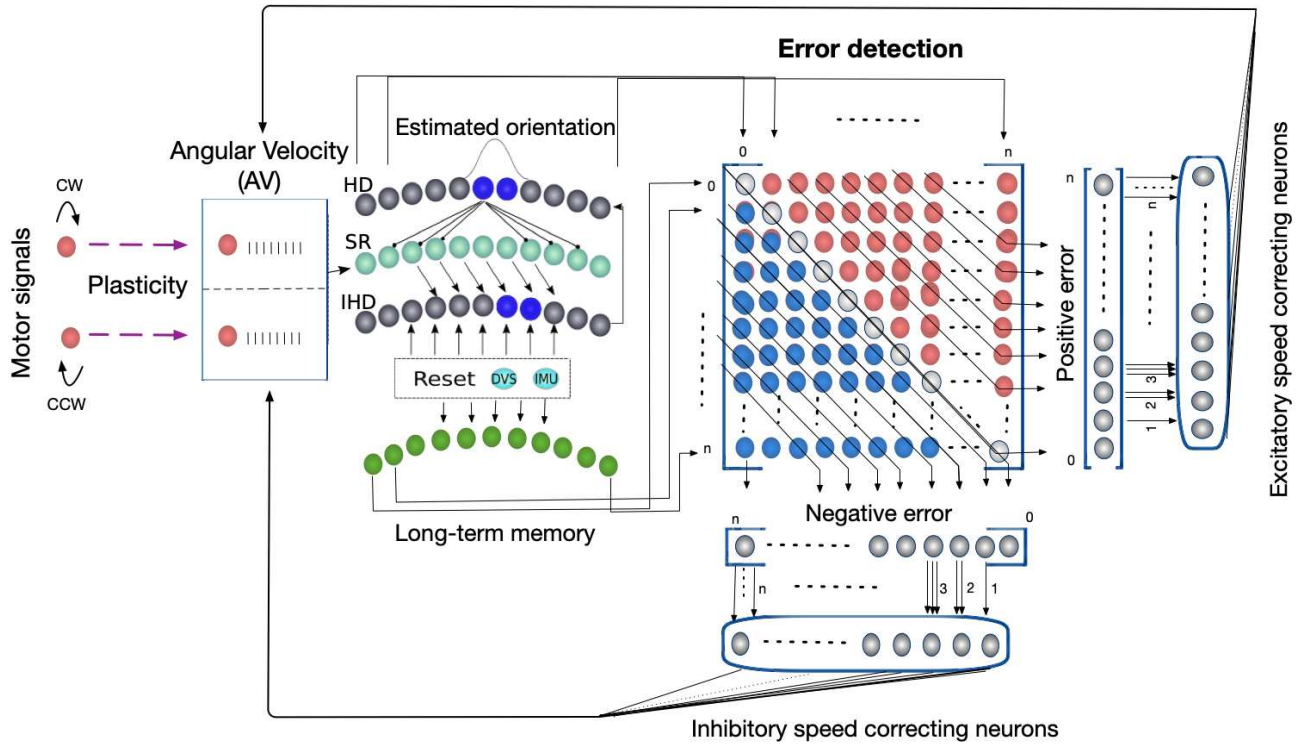

Figure 1: Connectivity scheme of the proposed SNN Architecture. The left part consists of the HD network, the right part of the error detection network that determines angular velocity rate learning.

increased.

In our network, the error estimating neurons recruit excitatory and inhibitory speed correcting neurons in order to change the firing rate of the AV neurons. The error estimating output neurons encode the error using space encoding (i.e. the identity of the most active neuron encodes the value of the error). This representation is converted to a rate code of the Speed Correcting neurons: each neuron in the positive or negative error population is connected to a certain number of the Speed Correcting neurons, proportional to the detected error magnitude (Fig. 1). E.g., the error neuron that represents the lowest error is connected to a single speed correcting neuron, while an error neuron that represents a high error is connected to a dozen of speed correcting neurons. AV neurons are driven by CW and CCW movement neurons. A weight change of the respective plastic synapses is induced by the activity of the Speed Correcting neurons. The weight is decreased upon the additional inhibitory input when a negative error is detected, or increased upon additional excitatory input during positive error detection. The synaptic weight determines the firing rate of the AV neurons, which drive the HD network (activating the shifting layers) and hence it influences the integration speed in the HD network (i.e. the movement speed of the bump).

2.3 Error-Driven Learning

Plastic synapses can be specified on the neuromorphic processor Loihi by setting a user-defined learning rule. We define plastic synapses that connect movement driven CW/CCW and AV neurons to follow a reinforcement-gated spiking rate based learning rule. Whenever the post-synaptic firing frequency is higher than the pre-synaptic frequency, the synaptic weight is more likely to increase. Likewise, whenever the pre-synaptic frequency is higher than the post-synaptic frequency, the synaptic weight is more likely to decrease. The post-synaptic frequency is modulated by the Speed Correcting neurons and a weight change is induced only if an error is detected and a reinforcement signal is received.

The weight update follows Eq. (1):

$$\Delta w = -r1 * x0 + r1 * (w_{max} - w) * (y1 * x0 + y0 * x1). \quad (1)$$

Here, r1 denotes the reward trace (representing the detected error), x0 denotes a presynaptic spike, and y1 denotes the postsynaptic trace (spikes convolved with an exponential temporal kernel). The maximal weight is limited by a value of w_{max}, which is 120 in our case (weight on Loihi can have values up to 256).

3 Results

3.1 Rate based velocity integration

We realized the neural architecture on Loihi using 32 neurons in each of the HD, IHD, and the shifting rings. We

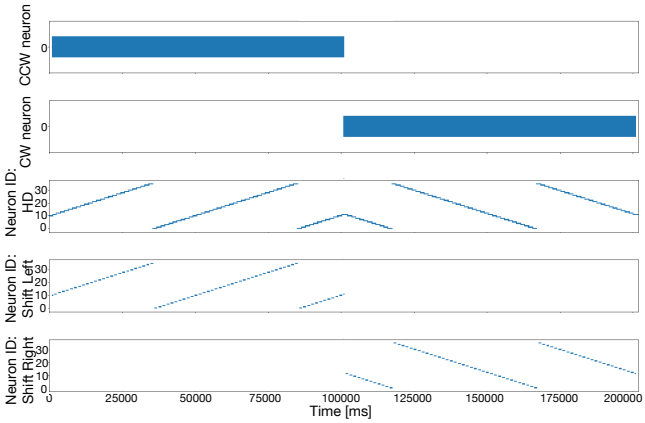

Figure 2: Neural activity during path integration. In the first half of the simulation, the CW neuron is active, in the second half the – CCW neuron initiates the HD neural activity to move in the other direction.

stimulated the CW and CCW neurons with spike generators that fire at different rates and recorded the activity in all sub-populations.

Fig. 2 shows the firing activity of the CW, CCW, HD, SR, SL and IHD neurons over a simulation of 30 seconds (at 1ms time step). The network first performs two full clockwise turns and then two full counter-clockwise turns, demonstrating that the HD network can indeed integrate velocity signals of different direction. Next, we recorded one full turn (starting at HD neuron 12) using different firing rates of the input (AV) neurons, see Fig. 3. The slope of the firing activity on the HD ring increases as a function of input firing rate. Fig. 4 shows the dependence of the angular velocity computed from the network activity (bump movement) on the firing rate of the AV neurons. Fig. 4 shows that the speed of the activity bump in the HD population monotonically increases with increasing input firing rate.

3.2 Learning the input rates using speed correcting neurons

In order to show how different firing rates can be learned by the network, we stimulate specific neurons on the ”correct” HD ring. Fig. 5 shows firing of the ”correct” HD and HD neurons, Error Encoding neurons, and the Speed Correcting pools over a simulation of 60 seconds in an example of positive error. Different neurons in the ”correct” HD population are briefly stimulated every 18 seconds and an error is computed by the network autonomously if the true and estimated orientation do not match.

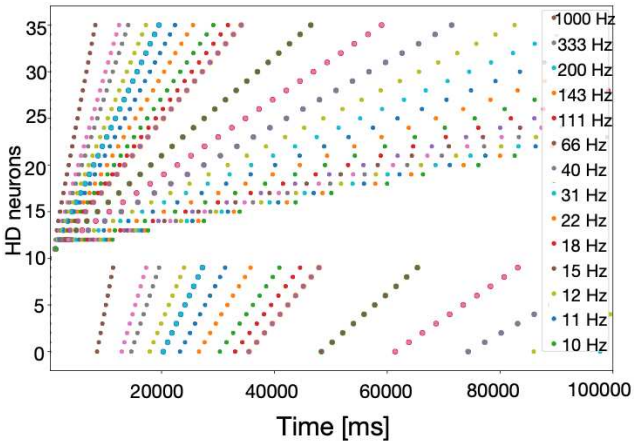

Figure 3: A full rotation estimated by the HD network for different firing rates of the input (AV) neurons. The input firing rate determines the slope of the activity drift and thus the speed of path integration.

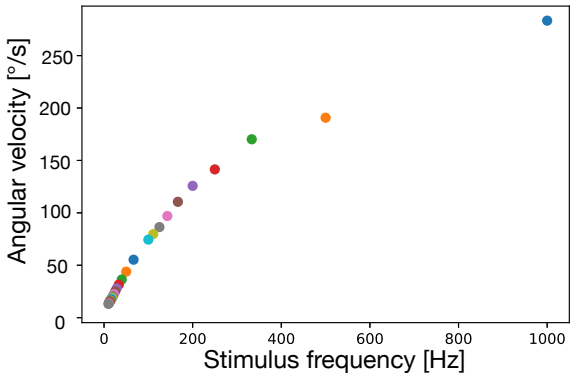

Figure 4: Dependence of represented angular velocity on the input firing frequency to the HD neuronal population.

In Fig. 5, the first active neuron in the ”correct” HD population is the same as the currently active HD neuron. Thus, the error is zero and learning does not occur. The next time a different true orientation is shown (at second 20), which corresponds to a positive error of 4, leading to 4 neurons in the excitatory Speed Correcting pool becoming active. The firing rate of the AV neurons is increased during learning. After the error signal has vanished, the increased synaptic weight maintains an increased firing rate of AV neurons. This can be seen in the effect on the slope of the raster plot of activity in the HD population (second from top plot in Fig. 5). When the true orientation is shown for the third time in this experiment (second 38), the computed error has a size of 15, hence activating 15 speed correcting neurons. Again, an increase in firing rate of the AV neurons leads to a higher slope of the HD neural activity movement.

The following experiment simulates a case when the true

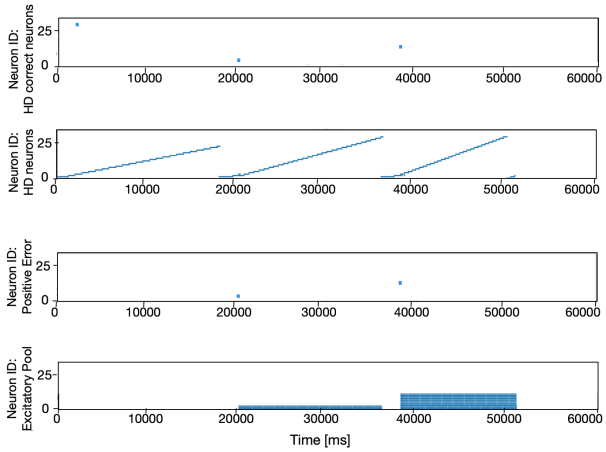

Figure 5: Firing activity during learning driven by a positive error. The true orientations is further ahead than the estimated orientation, signaling the network to increase the input rate which is controlling the shift in the HD population.

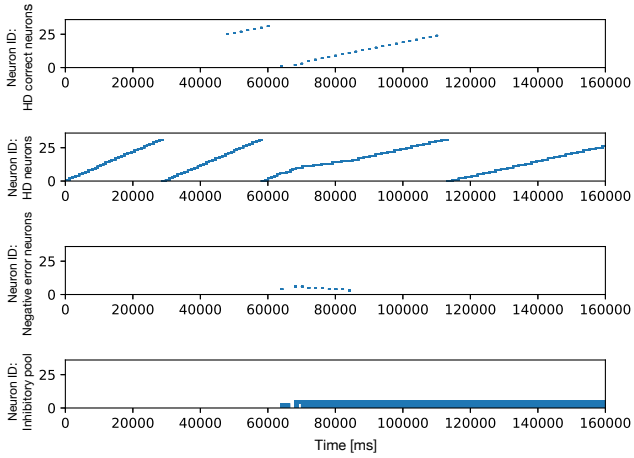

Figure 6: Continuously showing the true orientation enables the network to match the speed of the moving activity bump in the HD population to the true angular velocity of the robot. Here, the true angular velocity (top) is slower than the initial speed of the activity bump in the HD population. However, after detecting a negative error (third row), the network learns to slow down the input firing rate by recruiting inhibitory speed correcting neurons (bottom). The learned velocity stays the same after information about the true heading is removed (after second 110).

orientation is known for a full turn. Fig. 6 shows neuronal activity recordings from the Loihi chip when the true orientation is known from 47 to 110 seconds and the "correct" HD population is continuously stimulated to represent this information. In Fig. 6, a set of negative error detecting neurons becomes active and changes the shifting speed of the neuronal activity in the HD population to match the true angular velocity of the robot.

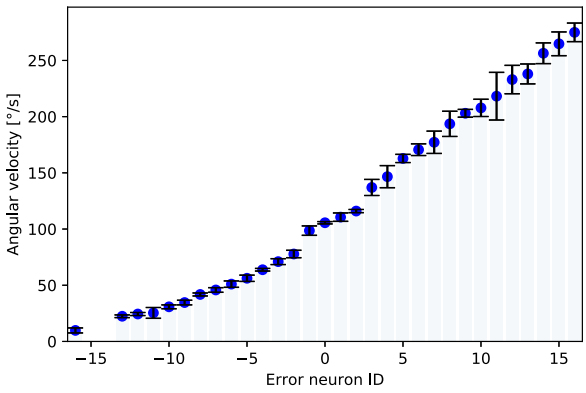

Figure 7: The learned angular velocity, represented by the speed of the activity bump in the HD population, driven by activity of different Error neurons. The initial speed of the activity bump in the HD corresponds to velocity 100 $circ/s$ (Error = 0). Different activated error neurons correspond to different simulated "correct" velocities. Dots show the mean and error bars indicate the standard deviation over 5 trials.

By stimulating the true HD neuron at different positions and recording the resulting path integration behavior on the HD ring, we evaluated how well different angular velocities can be learned. We performed five trials of simulation with each of the presented true velocities. Fig. 7 shows the mean and standard deviation of the angular velocity, represented by the speed of the activity bump in the HD population after learning, depending on the magnitude of the induced error (represented by the identity of the active neuron in the Error estimating population output layers). We can see that the angular velocities from 8 up to 286 degree per second can be realized by the network, while the standard deviation of the learned speed increases slightly for positive errors.

3.3 Convergence with a single visual landmark

In order to evaluate the system's learning performance over several encounters with the same landmark, we recorded activity during 10 simulated turns of the robot observing a single visual landmark. Landmark detection activates the corresponding neuron in the "correct" HD population and an error is estimated accordingly. The path integration network slowly adjusts its integration speed by decreasing the weights of plastic synapses.

Fig. 8 shows how the angular velocities, determined from the network's activity, change over time in this experiment. Different colors correspond to trials with different initial

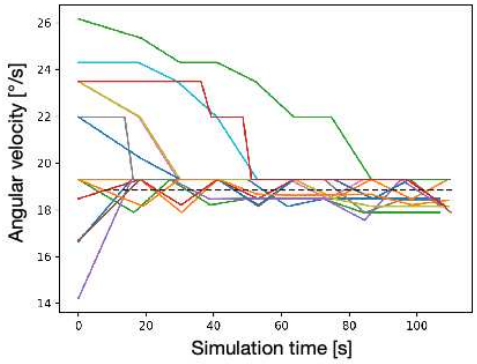

Figure 8: Approaching the true angular velocity (black dashed line) starting with 14 different initial neural input rates (shown by different colors). These firing rates determine the neural network's integration speed (speed of movement of the activity bump in the HD population), i.e. its estimated angular velocity. Simulation time corresponds to 10 full rotations in which a landmark is simulated at a single orientation. In all trials, the angular velocity of the neural network approaches the ground truth angular velocity.

neural angular velocities. The black dashed line corresponds to the true simulated angular velocity. In all trials, starting with different speeds, the neural network brings the speed of the moving bump to correspond to the true angular velocity.

4 Conclusion

We developed an SNN architecture for 1D path integration (orientation estimation based on motor commands) on a neuromorphic device Loihi. The neuronal architecture autonomously learns to match the internal representation of the robot's angular velocity (realized as the speed of movement of an activity bump) to the actual angular velocity of the robot. Learning can proceed based on both continual and intermittent visual feedback about the correct orientation of the robot. The network can continuously fine-tune the speed with which neuronal activity is shifted based on the external cues about the correct pose – a critical property for reliable and reproducible performance in real-world situations. The network structure is inspired by biological findings on navigational cell types rodents and the head direction network of the fruit fly. This work presents a first architecture for error estimation and autonomous, online learning for calibration of a path integration system in an SNN, realized in neuromorphic hardware.

Acknowledgments

This work was supported by SNSF Grant PZOOP2_168183(Ambizione) and ZNZ Fellowship.

References

Benjamin, B. V., et al., Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations, *Proceedings of the IEEE*, *102*(5), 699–716, doi:10.1109/JPROC.2014.2313565, 2014.

Cheung, A., Estimating Location without External Cues, *PLoS Computational Biology*, *10*(10), doi:10.1371/journal.pcbi.1003927, 2014.

Cheung, A., D. Ball, M. Milford, G. Wyeth, and J. Wiles, Maintaining a Cognitive Map in Darkness: The Need to Fuse Boundary Knowledge with Path Integration, *PLoS Computational Biology*, *8*(8), doi:10.1371/journal.pcbi.1002651, 2012.

Chicca, E., F. Stefanini, C. Bartolozzi, and G. Indiveri, Neuromorphic electronic circuits for building autonomous cognitive systems, *Proceedings of the IEEE*, *102*(9), 1367–1388, doi:10.1109/JPROC.2014.2313954, 2014.

Davies, M., et al., Loihi: A Neuromorphic Manycore Processor with On-Chip Learning, *IEEE Micro*, doi:10.1109/MM.2018.112130359, 2018.

Indiveri, G., E. Chicca, and R. J. Douglas, Artificial cognitive systems: From VLSI networks of spiking neurons to neuromorphic cognition, *Cognitive Computation*, *1*(2), 119–127, doi:10.1007/s12559-008-9003-6, 2009.

Kreiser, R., M. Cartiglia, J. N. Martel, J. Conradt, and Y. Sandamirskaya, A Neuromorphic Approach to Path Integration: A Head-Direction Spiking Neural Network with Vision-driven Reset, in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, doi:10.1109/ISCAS.2018.8351509, 2018.

McNaughton, B. L., F. P. Battaglia, O. Jensen, E. I. Moser, and M.-B. Moser, Path integration and the neural basis of the 'cognitive map', *Nature Reviews Neuroscience*, *7*, 663, 2006.

Milford, M. J., G. F. Wyeth, and D. Prasser, RatSLAM: A Hippocampal Model for Simultaneous Localization and Mapping, *Proceeding of the 2004 IEEE international Conference on Robotics & Automation*, pp. 403–408, doi:10.1109/ROBOT.2004.1307183, 2004.

Seelig, J. D., and V. Jayaraman, Neural dynamics for landmark orientation and angular path integration, *Nature*, *521*(7551), 186–191, doi:10.1038/nature14446, 2015.

Skaggs, W. E., J. J. Knierim, H. S. Kudrimoti, and B. L. McNaughton, A model of the neural basis of the rat's sense of direction, in *Advances in neural information processing systems*, pp. 173–180, 1995.

Tang, H., R. Yan, and K. C. Tan, Cognitive Navigation by Neuro-Inspired Localization, Mapping, and Episodic Memory, *IEEE Transactions on Cognitive and Developmental Systems*, doi:10.1109/TCDS.2017.2776965, 2018.

Thakur, C. S., et al., Large-scale neuromorphic spiking array processors: A quest to mimic the brain, *arXiv preprint arXiv:1805.08932*, 2018.

Turner-Evans, D., S. Wegener, H. Rouault, R. Franconville, T. Wolff, J. D. Seelig, S. Druckmann, and V. Jayaraman, Angular velocity integration in a fly heading circuit, *Elife*, *6*, e23,496, 2017.